Car 1451, Where are You?

SILICON UPDATE

Tom Cantrell

A Look at the IEEE 1451 Standard

Smart sensors aren't new, but the IEEE 1451 standard is. As Tom investigates this new development, he asks whether we're entering the digital age of sensors. Should you scrap all of your op-amp databooks?

Although the concept of smart sensors isn't new, as of today, the traditional signal chain, composed of sensor, signal conditioning, and A/D conversion, is proving stubbornly resistant to change. This old soldier not only isn't dying but doesn't seem to be fading away, either.

Not that there haven't been some inroads. For instance, various devices have emerged that combine a number of links in a single device, such as micromachined pressure sensors and accelerometers that incorporate signal conditioning to present a high-level (e.g., 0–5 V) output. Even brainier sensors (common examples are temperature and optical) throw in the ADC as well, delivering a digitized output (parallel I/O or pulse train) for direct connection to your favorite micro or DSP.

That's all well and good, but problems remain. For instance, there's little agreement on the digital I/O format and even less on the software machinations required to get at our beloved 1s and 0s. The result is that a seemingly simple change from brand *x* to brand *y* is likely to require a wholesale redesign of both hardware and software.

And it gets worse. So far, we're only talking about basic point-to-point lashups, which work fine for a few channels. But, what if the goal is to minimize wiring by connecting multiple devices on a single bus? Of course, the analog world has long relied on the good old 4–20-mA current loop for just such a purpose.

Ah, but the digital world is another story. Talk about putting multiple digital sensors on a single bus, and you open a closet door behind which rattle a bewildering array of skeletons. In fact, the so-called fieldbus concept is so confusing that, as far as I can tell, the whole idea threatens to collapse under its own weight.

Consider the sidebar "Sensor Control Networks," which shows you the latest (but likely not the last) version of the line card that confronts designers. Can anybody out there make sense of this, much less pick the likely winners and losers?

THE GREAT BYTE HOPE

Lurking at the edge of the ring is a new contender, the IEEE 1451 standard,

Line	Logic	Driven By	Function
DIN	Positive logic	NCAP	Address and data from transport from NCAP to STIM
DOUT	Positive logic	STIM	Data transport from STIM to NCAP
DCLK	Positive logic	NCAP	Positive-going edge latches data on DIN and DOUT
NIOE	Active low	NCAP	Signals that the data transport is active and delimits data transport framing
NTRIG	Negative logic	NCAP	Performs triggering function
NACK	Negative logic	STIM	Serves two functions: trigger acknowledge and data transport acknowledge
NINT	Negative logic	STIM	Used by the STIM to request service from the NCAP
NSDET	Active low	STIM	Used by the NCAP to detect the presence of a STIM
POWER	N/A	NCAP	Nominal 5-V power supply
COMMON	N/A	NCAP	Signal common or ground

Table 1—*The 1451.2 standard defines what is known as the TII, a 10-pin clocked serial transducer-independent interface.*

that purports to bring sensors into the digital age and then some.

I'm usually rather skeptical whenever I hear about the latest and greatest ultimate solution that's going to sweep all problems and uncertainties into a tidy little package for disposal on the ash heap of computer history. Heard it before and will hear it again, although the next big thing almost never lives up to its advanced billing.

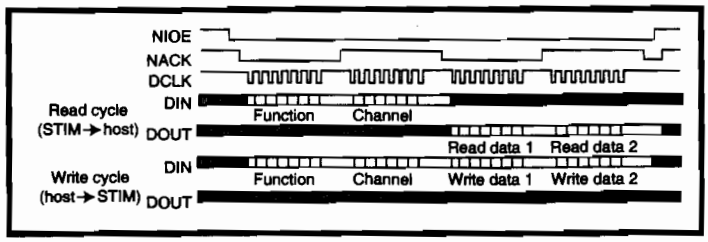

Figure 1—*The TII features a data clock (DCLK), separate data lines (DIN and DOUT), various control signals, and power and ground. NACK is used to throttle transfers at the byte level, eliminating the need for especially fast hardware to prevent overrun.*

Thus, for now at least, I'm not recommending that anyone chuck their op-amp databooks. Judgment certainly needs to be withheld until you understand what's under the hood and, even further, satisfy yourselves that the wind is blowing in a favorable direction.

However, I will say this for 1451. First, as an IEEE standard, you're given some assurance that it's both open and credible. Yes, I say "some" rather than "total" because we're all familiar with the way special interests sometimes jockey for position behind the standards veneer, attempting to gain proprietary advantage.

As for credibility, heck, even the long-buried S-100 bus got blessed by IEEE. But, I don't think 1451 has either been hijacked or is undeserving.

Another positive factor for 1451 is the relative absence of hype that often surrounds proposed standards. In fact, the effort could use a shot of PR. Few beyond those who specialize in sensors have even heard of it, much less know what it is. Compared to the puffery that often accompanies standards efforts, it appears 1451 proponents are adhering to a "walk softly but carry a big spec" strategy.

Enough background. Let's take a look at some of the details and you can judge for yourself. It's important to be informed since, à la democracy, you're going to get the smart sensors you deserve.

DOTS NOT ALL

The 1451 standard consists of four parts, "dot-1" to "dot-4" (i.e., 1451.1–1451.4), respectively. However, the history of 1451 is a little more convoluted than the nice sequential numbering scheme might indicate.

Development started a few years ago within IEEE and NIST on "A Smart Transducer Interface for Sensors and Actuators," now delivered under the auspices of IEEE 1451.2. Essentially, the standard defines what's known as a smart transducer interface module (STIM).

The key definitions incorporated into 1451.2 are a standard digital interface known as the transducer-independent interface (TII) and the format of a built-in transducer electronic datasheet (TEDS).

As you can see from Table 1, the electrical connection (i.e., TII) is relatively straightforward. It's a clocked serial interface (à la SPI) with most of the action revolving around a data clock (DCLK) and unidirectional data lines (DIN and DOUT).

NIOE is an output enable driven by the host (in 1451-speak, a network-capable application processor [NCAP]) that frames activity on the data lines. NACK is driven by the STIM to indicate that a byte transfer can proceed—that is, the host can drive DCLK.

Using NACK as a byte handshake reduces the timing burden on the STIM. The timing spec, which is shown in Figure 1, calls for all devices involved to support a minimum 6-kHz DCLK, but they're allowed to mutually agree on a higher rate.

Now, 6 kHz is well within the means of most chips' clock serial ports, so that's not a problem during a single-byte transfer. But, allowing only 166 ms between bytes might be problematic, depending on the amount of work the STIM needs to do between one byte transfer and the next. Requiring the host to hold off DCLK until the STIM gives the OK with NACK eliminates such concerns.

NTRIG can be asserted by the host to initiate a particular operation in the STIM. This action lets you establish precise timing when necessary, independent of any latency or uncertainty associated with communication and setup.

For the most part, all activities are carried out under direction of the host, except that NINT can be driven by the STIM to allow it to request service asynchronously. However, even in this case, the host response timing is not restricted. Thus, the host is in charge of its own destiny, which means that practically any device, fast or slow, can fill the role.

NSDET is also driven by the STIM as a way to signal its presence or absence. In simple configurations, it might be connected to ground on the STIM and pulled up on the host. But in other cases, the STIM may exercise explicit control to simulate a disconnect/reconnect (i.e., hot swap) cycle.

Finally, the host is responsible for providing power (up to 75 mA at 5 V) to the STIM. The STIM is certainly allowed to use an independent power source (e.g., if a high-power transducer or actuator is incorporated), but power for the interface itself should come

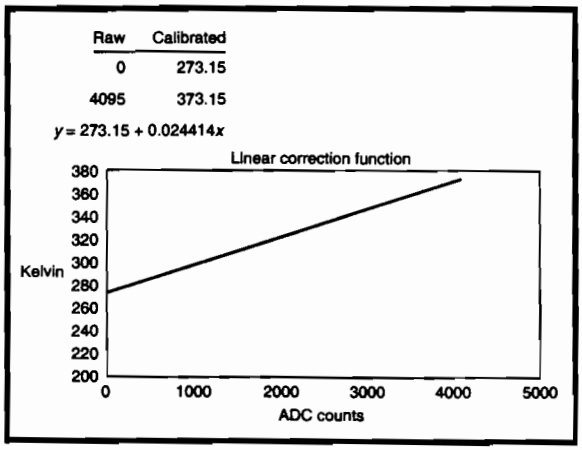

Figure 2—*1451.2 features an optional correction engine that handles calibration and standard units conversion. Here, it uses a polynomial equation to transform raw sensor output into meaningful information.*

from the host. This setup minimizes noise and ground loops.

One interesting note is the connector standard—to wit, there isn't one. Many of the early units incorporate a 10-pin (2 × 5) header, whereas others use a DA-15. In essence, the parties involved decided that, given the breadth of possible applications, there's more to be gained leaving users some flexibility than by dressing a particular connector.

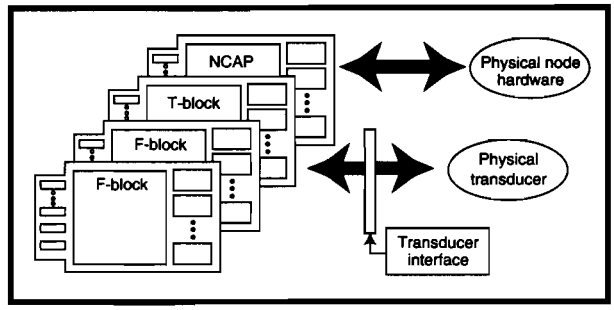

Figure 3—*The 1451.1 standard defines a software backplane for high-level objects, including NCAPs (Network Capable Application Processors), transducers (T-blocks), and functions (F-blocks).*

WHO'S ON FIRST?

The other major aspect of 1451.2 is its requirement that the STIM be the repository for the TEDS. This datasheet permits the host to find out what's on the other end of the wire—a sensors version of plug and play, if you will.

The TEDS comprises eight fields—two mandatory and six machine readable, each consisting of a length (byte count), data, and checksum. The two mandatory fields (Meta-TEDS and Channel TEDS) are machine-readable descriptions that uniquely identify the STIM and characterize the function of each channel.

The first optional field is the machine-readable Calibration TEDS. It relates to another major 1451.2 capability—the correction engine. In short, the idea is that the STIM should handle all adjustments and conversions needed to convert sensor input into something usable by the host.

Typically, correcting a sensor output involves using an equation to adjust a raw A/D reading. The 1451.2 approach supports a variety of transformations including linear, polynomial (see Figure 2), and multisegment polynomial.

The correction engine also accommodates conversion to and from IEEE 754 floating-point representations of SI (Standard International) units. STIMs from different vendors and based on completely different underlying technology can deliver exactly the same information to your program. Neat!

The next three fields are optional human-readable equivalents of the first three. Because humans come in a variety of flavors, a wide variety of languages and character sets is allowed, not just English and ASCII.

Another optional field is the Application-Specific TEDS, which is any (human readable) stuff you care to add. Finally, the spec makes provision for future growth with an arbitrary number of Extension TEDS.

The only definition associated with Extensions is an ID number that requires blessing by the IEEE. One ID number is reserved specifically to enable prototyping and experimentation prior to official assignment.

An interesting aside: Nothing says a STIM has to be a monolithic device. It's quite often the case that the sensor itself must work in an environment far too harsh for any chip.

Thus, you're free to connect the brains of the STIM and the sensor any way you please. However, although by no means enforceable, keep in mind that it's the intent of the spec that a sensor and its associated TEDS remain together until death do they part.

HANDS ON

As I mentioned, until now, 1451 action has largely been confined to sensor-industry gurus and insiders. Short of ordering the spec from the IEEE, there's been little the average embedded-system designer could do to check it out, much less get a head start on development.

I'm pleased to report that one outfit, Electronics Development Corporation (EDC), has stepped into the breach with a lineup of low-cost evaluation and development gear that puts the spec within reach of mere mortals.

In fact, EDC was an outsider until commissioned to help put together a 1451 demo at a tradeshow. They did this by tweaking their nifty CogniSense module, shown in Photo 1a, combining a PIC and signal-conditioning ASIC.

Subsequently, they jumped in with both feet and are certainly the best source I've found to help you get up to speed. Probably the best way to start is with their EDC 1451.2-KA Smart Transducer Interface kit (see Photo 1b).

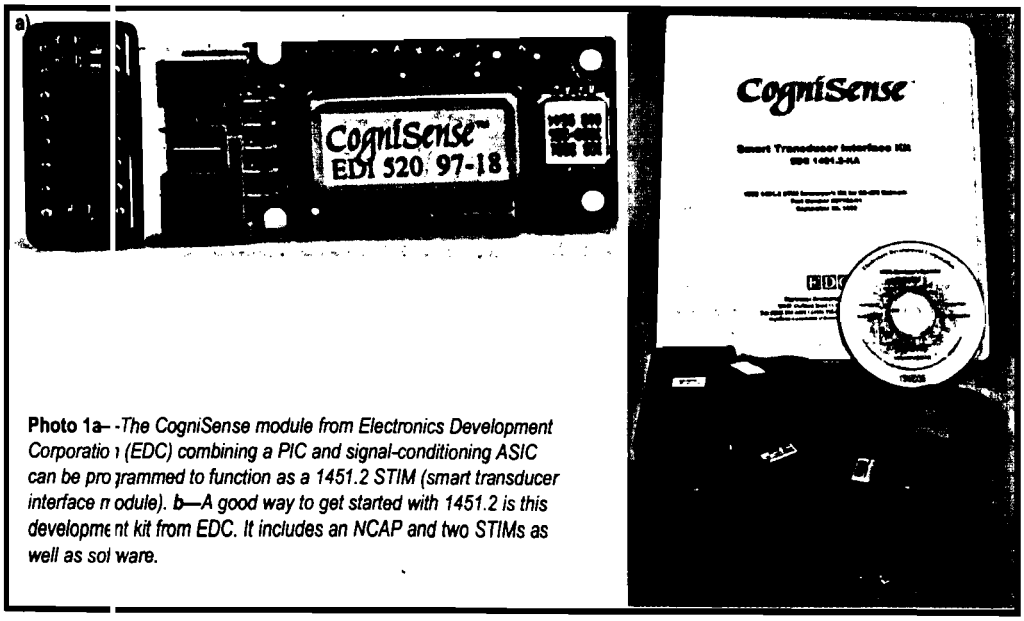

Photo 1a—*The CogniSense module from Electronics Development Corporation (EDC) combining a PIC and signal-conditioning ASIC can be programmed to function as a 1451.2 STIM (smart transducer interface module).* **b**—*A good way to get started with 1451.2 is this development kit from EDC. It includes an NCAP and two STIMs as well as software.*

Sensor Control Networks

This list (see http://129.6.36.211/Home/P1451/IeeeSite/contnet.htm) highlights the dilemma designers face. So many networks, so little time.

ARCNet—Attached Resource Computer Network, developed in 1977 by Datapoint. Maximum data rate of 2.5 Mbps.

ASI—Actuator Sensor Interface, developed in Germany by a consortium of sensor suppliers. A low-cost, bit-level system designed to handle four bits per message for binary devices in a master/slave structure operating over distances up to 100 m.

BACnet—Building Automation Control Network, an American Association of Heating, Ventilation, Refrigeration, and Air Conditioning Engineers (ASHRAE) standard developed by HVAC system suppliers. It supports networking options: ARCNet, Ethernet, a master/slave token passing (MS/TP) network based on RS-485 protocol, and LonWorks.

Bitbus—developed in 1984 by Intel around the 8044 microprocessor. Features multitasking with a master/slave structure using RS-485 serial linking.

CAN bus—Control Area Network bus, developed in Germany by Robert Bosch GmbH with Intel and Philips in the early '80s for automotive in-vehicle networking. This peer-to-peer Carrier Sense Multiple Access (CSMA) system supports selectable data transfer rates up to 1 Mbps and twisted-pair, fiber, coax, and RF media. CAN is ISO Standard 11898, approved for passenger-vehicle applications. CAN-based systems were approved by SAE as Standard J1850 for American passenger cars and Standard J1939 for trucks and large vehicles. A CAN in Automation (CIA) Group has been formed in Germany to work on application issues.

CEBus—originally approved as Consumer Electronics Bus by the Electronic Industries Association. Today, CEBus goes far beyond consumer electronics. Primarily used in home automation, it supports coax, RF, power-line, twisted pair, and infrared and has provisions for fiber-optics media.

DeviceNet—version of CAN developed by Allen-Bradley. It features object-oriented software and is used in industrial control systems. It uses a four-wire (signal pair and power pair) shielded cable and supports up to 64 nodes per network at speeds up to 500 kbps at 100 m and 125 kbps at 500 m. An Open DeviceNet Vendors Association (ODVA) exists.

Foundation Fieldbus—formed from the merging of components of specifications by WorldFIP and Profibus supporters. It was formed to test and demonstrate fieldbus components to support an eventual single, universal fieldbus standard.

GPIB—General-Purpose Interface Bus, became the IEEE-488 Standard in 1978. It's more of a data-acquisition system with limited node capabilities and is used in laboratories and industrial instrument systems.

HART—Highway Addressable Remote Transducer, a network produced by Rosemount. It provides two-way digital communication atop traditional 4-20-mA loops. A HART organization has been formed.

Interbus S—open system developed by Phoenix Contact.This fast sensor/actuator data-ring-type bus uses RS-422 transceiver technology and handles analog via separate I/O modules. Up to 256 drops per network and up to 4096 digital I/Os can be supported. The network is deterministic with data throughput in the low milliseconds.

ISA SP50—organized in 1985 to develop a digital signal-based standard to complement the traditional 4–20-mA standard of the process industries. Beset by individual company interests as well as Profibus/WorldFIP polarizations, SP50 has had a long, tough trail in pursuing an acceptable fieldbus standard. But, recent progress is encouraging with the support

(continued)

For $450 you get an RS-485 network adapter (i.e., NCAP), RS-232–to–RS-485 adapter cable (connects the network adapter to a PC COM port), two Cogni-Sense-based STIMs (one preconfigured with an accelerometer and one for experimenting with your own sensor) plus TEDS editing and STIM control/viewing software. Up to 255 of the network adapters can be daisy-chained on the '485 bus, and they, along with the STIMs, are also sold separately.

THE BIGGER PICTURE

Although you may be just getting your first good look at 1451.2, as an approved IEEE draft standard, it's well on its way to hitting the street. However, dot-2 isn't the whole story.

What about dot-1, dot-3, and dot-4? Is it the infamous spec creep in which every Johnny-come-lately starts loading stuff on to what might otherwise be a good (i.e., simple) idea?

A key point is that the numbering isn't tied to chronology. IEEE 1451.1, although it has a working group in place, is not as far along as 1451.2. And, IEEE 1451.3 and 1451.4 are only at the PAR (Project Authorization Request) stage.

Rather, the numbering is meant to reflect a hierarchical level of abstraction, from a 1s-and-0s network on the dot-1 end to a mostly analog gadget on the other side of dot-4. Here's the (by necessity, very brief at this point) story.

Hewlett-Packard has been a driving force behind dot-1, which can be considered both a theoretical and pragmatic response to the fieldbus chaos reflected in the sidebar.

On the theoretical side, the benefit of 1451.2 compatibility will be hobbled by network incompatibilities. What good is a compatible sensor if it is hidden behind an incompatible network? Software developers still face that prospect hacking away at their programs to support different networks.

As well, HP is of the belief that Ethernet, while never designed for this purpose, has a lot going for it as a fieldbus—namely, it's cheap, it works, and it's widely available. But, simply adding Ethernet to the laundry list of sensor networking schemes will do little to further their cause.

(continued)

of the Fieldbus Foundation. Profibus and WorldFIP offer eventual migration paths to any forthcoming IEC 1158-SP50 world standard.

J1850—an SAE Standard for passenger cars covering mid-speed data rates optimized at 10.4 and 41.6 kbps—rates used by GM and Ford.

LonWorks—Local Operating Network, a distributed control network developed by Echelon. It uses custom Neuron chips implementing ISO/OSI seven-layer stack protocol. It supports media like twisted pair, coax, fiber optic, RF, infrared, and power line with data rates up to 1.25 Mbps for distances up to 500 m and 78 kbps at 2000 m.

Profibus—Process Field Bus, developed in Germany and strongly supported by Siemens. It is German DIN Standard 19245. Parts 1 and 2 are designated Profibus-FMS and cover automation in general. Part 3, Profibus-DP, is a faster system for factory automation. A fourth Profibus-PA part is in preparation for process control. A number of installations are operating covering various industries. Chips and tools are available.

SDS—Smart Distributed System, developed by Honeywell MicroSwitch. This open CAN-based system uses a four-wire cable (two twisted pairs; signal and power). It supports up to 128 nodes at speeds up to 1.25 Mbps interfacing with PLCs and PCs for industrial control applications.

Sercos—a bus developed in Europe for motors and motion-control applications.

Seriplex—developed by Automation Process Control (APC) Company. This ASIC-based multiplexing system offers both peer-to-peer and master/slave communications.

WorldFIP—Factory Information Protocol, a French National Fieldbus Standard based on the three OSI control-related layers 1, 2, and 7. There are a number of installations primarily in France and Italy. Chips and products are available. FICOMP (Fieldbus Consortium), begun in late 1992, is developing board-level products and software in accordance with IEC, Fieldbus Foundation, and WorldFIP specs.

Thus, 1451.1 formalizes what any good programmer would do when faced with such a circumstance: add a level of abstraction like an object-oriented API (application program interface) between the network host and the NCAP(s) that insulates the highest-level software from the details of exactly which hardware is being used.

The concept is that a system comprises a Lego block–like collection of objects: NCAP's, transducers, and functions that can publish and subscribe data (see Figure 3).

Meanwhile, 1451.3 accommodates networking on a smaller scale, addressing applications that call for an array of transducers in close proximity to the STIM, where a big network would be overkill. Basically, dot-3 exists to overcome the technicality in dot-2 that tightly couples a TEDS with a single transducer.

Finally, IEEE 1451.4 recognizes that there's a lot of analog know-how and installed base that won't disappear overnight. By defining a mixed mode transducer interface, it plans to support transmission of basic-TEDS digital information over an analog link.

The good news is that each higher numbered spec is completely contained within the confines of a lower numbered one. For instance, 1451.1 only sees 1451.2 STIMs regardless of whether they're monolithic devices or composed of a combination of 1451.3 and/or 1451.4 devices (see Figure 4).

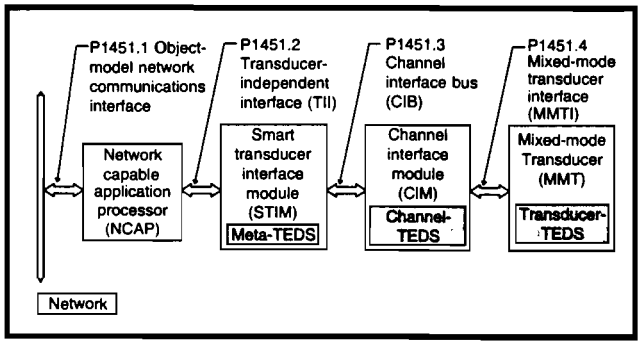

Figure 4—*Putting it all together, IEEE 1451 defines the signal chain from analog sensor to digital network.*

HIGH-SPEED PURSUIT

For standards that purport to move sensors into the digital age, I must say that the backers haven't done enough to bring us digital types onboard. For instance, although a number of papers have been presented at Sensors Expo over the years, there's been nary a one at the Embedded Systems Conference.

Message to 1451 folks: Don't forget those who design the systems and write the software that ultimately uses, and pays for, smart sensors.

Admittedly, I've only mentioned 1451 a couple of times in previous columns myself. I hope this column serves as a worthy first step, but more investigation and hands-on scrutiny is called for. Success of the standard certainly isn't guaranteed at this point, but if it does take off, it's going to be a big deal.

Now that we know Car 1451 is out there somewhere, let's find it.

Tom Cantrell has been working on chip, board, and systems design and marketing in Silicon Valley for more than ten years. You may reach him by E-mail at tom.cantrell@circuitcellar.com, by telephone at (510) 657-0264, or by fax at (510) 657-5441.

REFERENCES

Electronics Development Corp., www.elecdev.com

Hewlett-Packard, www.hpie.com, www.hpl.hp.com/techreports/98/HPL-98-166.html

http://129.6.36.211/Home/P1451/ieee.htm

http://129.6.36.211/Home/P1451/docs/nist.htm

http://129.6.36.211/Home/P1451/IeeeSite/P1451.htm

www.ic.ornl.gov/p1451/p1451.html